Hamburger Transplantation: Lightweight Additive Capability Enhancement via Neural Organ Insertion with Stitching Layers

Tatsuru Okada
Independent Researcher

Abstract

I present **Hamburger Transplantation**, a lightweight method for enhancing large language model (LLM) capabilities by additively inserting frozen MLP layers ("organs") extracted from specialist models, connected via thin learnable stitching layers. Unlike conventional fine-tuning or LoRA, my approach requires **only 30 seconds of training on a consumer laptop** (Apple M-series, no GPU cluster required), uses merely 200 samples of general-purpose text, and preserves the base model's original capabilities through residual connections. I demonstrate that inserting a reasoning organ (from DeepSeek-R1) and a coding organ (from Qwen2.5-Coder) into a Qwen2.5-7B host model improves HumanEval performance from 80.0% to **84.0%** (+4pp, prompt-independent). Extending to a triple-organ configuration with a Japanese language organ, I discover a systematic **Confidence Bias** problem: organ insertion suppresses neutral predictions in natural language inference tasks, collapsing a 3-class distribution into a 2-class one. I address this through **Assistant Axis Constraint Training**, a novel two-pass training procedure that penalizes activation deviation along the model's identity-encoding direction, restoring balanced predictions. Through comprehensive cross-scale experiments transplanting Japanese organs from models up to 2× the host's hidden dimension (7168 vs. 3584), I demonstrate that stitching layers enable **cross-architecture organ transplantation** with comparable perplexity (5.91–6.26 vs. 5.45 base). A systematic ablation over constraint strength (λ) and layer range reveals a sweet spot at λ=0.01, L=24–27, achieving **72.0%** on JNLI—surpassing the 68.0% base model while maintaining balanced class predictions. Cross-benchmark evaluation across five configurations (MMLU-STEM, JCommonsenseQA, MBPP) reveals that organ insertion enhances target capabilities (+8pp MBPP for code, +10pp JCommonsenseQA for Japanese) but introduces *soft interference* that degrades host knowledge (−7.5pp MMLU-STEM), even with frozen base parameters. The axis constraint recovers host knowledge but suppresses organ gains, exposing a fundamental tension in additive transplantation. All experiments are conducted on a single MacBook Pro (M4 Pro, 24GB).

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across diverse tasks, yet enhancing specific abilities—such as mathematical reasoning or code generation—typically requires expensive fine-tuning procedures. Parameter-efficient methods like LoRA Hu et al. [2022] reduce computational requirements but still demand task-specific datasets, GPU resources, and careful hyperparameter tuning. Full fine-tuning risks catastrophic forgetting of existing capabilities Kirkpatrick et al. [2017].

Recent work on Neural Organ Transplantation (NOT) Al-Zuraiqi [2026] demonstrated that MLP layers can be extracted from specialist models and transplanted into host models, with stitching

layers bridging representational gaps. However, the NOT approach focuses primarily on *layer replacement*—substituting host layers with donor layers—which risks degrading the host's original capabilities.

I propose **Hamburger Transplantation**, a fundamentally different approach based on *additive insertion*. Rather than replacing layers, I insert additional organ layers between existing host layers, connected via residual connections:

$$\mathbf{h}' = \mathbf{h} + \alpha \cdot f_{\text{stitch-out}}(\text{MLP}_{\text{organ}}(f_{\text{stitch-in}}(\mathbf{h}))) \tag{1}$$

where α is a learnable scale parameter initialized to 0.1, ensuring the organ's contribution begins small and the host model's behavior is preserved.

The name "Hamburger" derives from the architectural metaphor: the organ layer is sandwiched between host layers like a patty between buns, with stitching layers serving as the condiments that bridge the gap.

My key contributions are:

1. **Additive Organ Insertion**: A residual-based approach that preserves base model capabilities while adding new ones, unlike destructive layer replacement (§3).

2. **Ultra-Lightweight Training**: Stitching layers require only **30 seconds** of training with 200 general-purpose text samples on a consumer laptop (Apple M4 Pro, 24GB RAM).

3. **Composable Multi-Organ Architecture**: Multiple organs can be stacked at different layers for combined capability enhancement, demonstrated with dual-organ (+4pp on HumanEval) and triple-organ configurations (§5.1).

4. **Discovery of Confidence Bias**: I identify a systematic failure mode where organ insertion suppresses neutral predictions in NLI tasks, collapsing a 3-class distribution into 2 classes (§5.3).

5. **Assistant Axis Constraint Training**: A novel two-pass training procedure that penalizes activation deviation along the model's identity direction, resolving Confidence Bias while preserving organ contributions (§3.6).

6. **Cross-Scale Transplantation**: Stitching layers enable transplantation from donors up to 2× the host's hidden dimension, with cross-architecture organs from InternLM and Yi families (§5.4).

7. **Constraint Ablation**: Systematic study of $\lambda\times$ layer-range interactions reveals a sweet spot (λ=0.01, L24–27) that surpasses the base model on JNLI (§5.6).

2 Related Work

2.1 Parameter-Efficient Fine-Tuning

LoRA Hu et al. [2022] and its variants (QLoRA Dettmers et al. [2023], DoRA Liu et al. [2024]) enable fine-tuning with reduced memory by learning low-rank weight updates. While effective, these methods require task-specific training data, typically thousands to millions of examples, and training times of hours to days on GPU hardware. My approach requires only 200 general-purpose samples and 30 seconds on a CPU/MPS device.

2.2 Model Merging and Composition

Model merging techniques Wortsman et al. [2022], Ilharco et al. [2023] combine weights from multiple fine-tuned models. Methods like TIES-Merging Yadav et al. [2023] and DARE Yu et al. [2024] address interference between merged parameters. Unlike weight merging, my approach preserves architectural separation between components, avoiding destructive interference.

2.3 Mixture of Experts

Mixture-of-Experts (MoE) architectures Shazeer et al. [2017], Fedus et al. [2022] route inputs to specialized sub-networks. While conceptually related, MoE models require training from scratch or expensive upcycling Komatsuzaki et al. [2023]. My organs are extracted post-hoc from pre-trained specialists and require minimal adaptation.

2.4 Neural Organ Transplantation

The NOT framework Al-Zuraiqi [2026] extracts functional "organs" (MLP layers) from donor models and transplants them into host models using stitching layers. My work differs in three key aspects: (1) I use *additive insertion* rather than layer replacement, preserving host capabilities; (2) I demonstrate multi-organ composition; and (3) I identify the critical single-epoch training constraint.

2.5 Model Stitching

Model Stitching Bansal et al. [2021], Csiszárik et al. [2021] uses linear transformations to connect representations from different models, providing insights into representational similarity. I adopt stitching layers as practical connectors between host and donor representations.

3 Method

3.1 Overview

Given a pre-trained host model $\mathcal{M}_H$ with L transformer layers, and one or more donor models $\{\mathcal{M}_{D_k}\}$, Hamburger Transplantation proceeds in three stages:

1. **Organ Extraction**: Extract MLP layers from donor models as portable "organs."
2. **Organ Insertion**: Insert organs between host layers with learnable stitching layers.
3. **Stitching Training**: Train only the stitching layers (0.3% of total parameters) for 1 epoch on general-purpose text.

3.2 Organ Extraction

From each donor model $\mathcal{M}_{D_k}$, I extract a single transformer MLP block consisting of the SwiGLU Shazeer [2020] components:

$$\text{Organ}_k = \{\mathbf{W}^{(k)}_{\text{gate}}, \mathbf{W}^{(k)}_{\text{up}}, \mathbf{W}^{(k)}_{\text{down}}\} \tag{2}$$

along with the associated layer normalization parameters and dimensional configuration. These weights are frozen and never modified during training.

3.3 Stitching Architecture

Each organ is wrapped with two learnable stitching layers and connected via a residual path:

$$\mathbf{h}' = \mathbf{h} + \alpha \cdot S_{\text{out}}\left(\text{MLP}_{\text{organ}}\left(\text{LN}\left(S_{\text{in}}(\mathbf{h})\right)\right)\right) \tag{3}$$

where:

- $S_{\text{in}} : \mathbb{R}^{d_H} \to \mathbb{R}^{d_D}$ is the input stitching layer (linear projection)
- $S_{\text{out}} : \mathbb{R}^{d_D} \to \mathbb{R}^{d_H}$ is the output stitching layer
- LN is layer normalization
- α is a learnable scale parameter initialized to 0.1
- d_H and d_D are the host and donor hidden dimensions

Initialization. When $d_H = d_D$ (same-family transplantation), stitching layers are initialized as identity matrices. When $d_H \neq d_D$ (cross-family), they are initialized with $\mathcal{N}(0, 0.02)$.

The residual connection is critical: it ensures that at initialization (before training), the organ's contribution is small ($\alpha = 0.1$), preserving the host model's behavior. This differs fundamentally from NOT's replacement approach.

3.4 Multi-Organ Composition

Multiple organs can be inserted at different positions in the host network:

$$\mathbf{h}'_l = \begin{cases} \mathbf{h}_l + \alpha_k \cdot \text{Organ}_k(\mathbf{h}_l) & \text{if layer } l \in \mathcal{P} \\ \mathbf{h}_l & \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{P}$ is the set of insertion points. Insertion positions were chosen to distribute organs across network depth, following the empirical finding that transformer layers encode progressively more abstract features—lower layers capture syntactic features while upper layers encode semantic representations Tenney et al. [2019], Jawahar et al. [2019]—and that middle-to-upper layers exhibit greater redundancy, making them more amenable to additive modification Men et al. [2024]. In my triple-organ experiments, I use:

- **Reasoning organ** (DeepSeek-R1) at layer 5 (early layers)
- **Coding organ** (Qwen2.5-Coder) at layer 11 (middle layers)
- **Japanese organ** (various donors) at layer 17 (late-middle layers)

3.5 Cross-Scale Transplantation

When the donor hidden dimension d_D differs from the host dimension d_H, the stitching layers $S_{\text{in}} : \mathbb{R}^{d_H} \to \mathbb{R}^{d_D}$ and $S_{\text{out}} : \mathbb{R}^{d_D} \to \mathbb{R}^{d_H}$ perform a non-trivial dimensional bridge. I evaluate donors with scale ratios from 1.0× to 2.0×:

- Qwen2.5-14B-Instruct: d_D=5120 (1.43×)
- Qwen2.5-32B-Instruct: d_D=5120 (1.43×, larger intermediate)
- InternLM2.5-20B-Chat: d_D=6144 (1.71×, different architecture family)
- Yi-1.5-34B-Chat: d_D=7168 (2.00×, different architecture family)

3.6 Assistant Axis Constraint Training

I discover that organ insertion introduces a *Confidence Bias*: the model's output distribution shifts systematically, suppressing uncertain (neutral) predictions in classification tasks (§5.3). To address this, I propose **Assistant Axis Constraint Training**, which constrains stitching layer training to preserve the model's identity-encoding direction.

Axis Extraction. Following Representation Engineering approaches, I extract the *assistant axis* $\mathbf{a} \in \mathbb{R}^{L \times d_H}$ by computing the difference in mean activations between the model's default response mode and diverse persona modes across 20 roles (consultant, analyst, poet, rebel, etc.) and 10 questions:

$$\mathbf{a}_l = \mathbb{E}_{\text{default}}[\mathbf{h}_l] - \mathbb{E}_{\text{persona}}[\mathbf{h}_l] \tag{5}$$

This axis encodes the direction in activation space that characterizes the model's "being itself"—its calibrated, balanced response behavior.

Constrained Training. I train stitching parameters with an augmented loss:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda \sum_{l \in \mathcal{C}} \left\| \text{proj}_{\hat{\mathbf{a}}_l}(\mathbf{h}_l^{\text{organ}} - \mathbf{h}_l^{\text{base}}) \right\|^2 \tag{6}$$

where $\hat{\mathbf{a}}_l = \mathbf{a}_l / \|\mathbf{a}_l\|$ is the normalized axis at layer l, $\mathbf{h}_l^{\text{organ}}$ and $\mathbf{h}_l^{\text{base}}$ are activations with and without organs at layer l, $\mathcal{C}$ is the set of constraint layers, and λ controls the constraint strength. This requires a two-pass forward: one with organs enabled, one with organs disabled (scale α temporarily set to 0).

3.7 Stitching Layer Training

Objective. I minimize the standard language modeling loss:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(x_t | x_{<t}; \theta_S, \theta_H, \theta_O) \tag{7}$$

where θ_S are the stitching parameters (trainable), θ_H are the host parameters (frozen), and θ_O are the organ parameters (frozen).

Training Protocol. The key finding of this work is that training should be minimal:

- **Data**: 200 samples from WikiText-2 (general-purpose text)
- **Epochs**: **Exactly 1** (see Section 5.7)
- **Optimizer**: AdamW with lr $= 10^{-3}$, weight decay $= 0.01$
- **Gradient clipping**: Max norm 1.0
- **Sequence length**: 128 tokens
- **Trainable parameters**: $\sim$25M per organ (0.3% of total)

Algorithm 1 Hamburger Transplantation

Require: Host model $\mathcal{M}_H$, Donor organs $\{O_k\}$, Insertion points $\{l_k\}$, Training data $\mathcal{D}$

Freeze all parameters of $\mathcal{M}_H$ and $\{O_k\}$
for each organ O_k **do**
  Initialize $S_{\text{in}}^{(k)}, S_{\text{out}}^{(k)}, \alpha_k$
  Register forward hook at layer l_k of $\mathcal{M}_H$
end for
$\theta_S \leftarrow \{S_{\text{in}}^{(k)}, S_{\text{out}}^{(k)}, \alpha_k\}_k$ {Trainable params only}
for each sample $x \in \mathcal{D}$ **do**
  Compute $\mathcal{L}(x; \theta_S)$ {Language modeling loss}
  Update θ_S via AdamW
end for
return Enhanced model $\mathcal{M}_H$ with organs

4 Experiments

4.1 Setup

Host Model. Qwen2.5-7B-Instruct Qwen Team [2024] with 28 transformer layers, hidden dimension 3584, loaded in float16.

Donor Models and Organs.

- **Math Organ**: MLP from layer 15 of DeepSeek-R1-Distill-Qwen-7B Guo et al. [2025] (hidden dim: 3584). Inserted after host layer 5.
- **Code Organ**: MLP from layer 15 of Qwen2.5-Coder-7B-Instruct Hui et al. [2024] (hidden dim: 3584). Inserted after host layer 11.

All models were obtained from HuggingFace Hub in January 2025. As model weights on HuggingFace may be updated without version changes, exact reproducibility requires the same checkpoint snapshots.

Hardware. All experiments were conducted on a MacBook Pro with Apple M4 Pro chip and 24GB unified memory. No external GPU or cloud compute was used.

Evaluation Benchmarks.

- **HumanEval** Chen et al. [2021]: 50 code generation problems (seed=42), greedy decoding.
- **JNLI** (from JGLUE Kurihara et al. [2022]): Japanese Natural Language Inference, 50 validation samples (seed=42). 3-class: entailment (E=9), contradiction (C=15), neutral (N=26).
- **JCommonsenseQA** (from JGLUE): Japanese commonsense QA, 50 samples (seed=42).
- **Perplexity**: Mixed English-Japanese corpus, 18 samples.

4.2 Baselines and Evaluation Conditions

1. **Base Model**: Qwen2.5-7B-Instruct without modification
2. **Direct Insertion (v1)**: Organ MLP inserted without stitching layers
3. **Dual Organ**: Math@5 + Code@11 with trained stitching

4. **Triple Organ**: Math@5 + Code@11 + Japanese@17 (various donors)

5. **Triple + Axis**: Triple organ with axis-constrained stitching training

Fair 4-Condition Comparison. For HumanEval, I employ a full factorial experiment: 2 models (Base / Dual Organ) × 2 prompts (Simple / Reasoning), all with `max_new_tokens=512` for fairness.

5 Results

5.1 Main Results: Code Generation (HumanEval)

Table 1 presents developmental results on a 10-problem subset, and Table 2 presents the main fair comparison on 50 problems.

Table 1: Developmental results (10-problem subset, seed=42).

Model	**Pass@1**	**Train Time**	**Train Data**	**HW**
Base Model (Qwen2.5-7B)	8/10 (80%)	—	—	—
Direct Insertion (v1)	2/10 (20%)	0	0	—
Single Organ (Code, 3ep)	5/10 (50%)	90s	500 code	MPS
Single Organ (Code, 1ep)	8/10 (80%)	30s	200 wiki	MPS

Table 2: Fair 4-condition comparison on HumanEval (50 problems, seed=42, `max_new_tokens`=512 unified). Training time: 60s on Apple M4 Pro.

	Simple Prompt	**Reasoning Prompt**	**Δ (Prompt)**
Base Model	40/50 (80.0%)	39/50 (78.0%)	−1
Dual Organ	**42/50 (84.0%)**	41/50 (82.0%)	−1
Δ (Organ)	**+2 (+4.0%)**	**+2 (+4.0%)**	

Effect decomposition. The organ effect is a consistent +2 problems (+4.0%) regardless of prompt type. The reasoning prompt ("Think step by step") actually decreases performance by 1 problem (−2.0%) for both models, indicating it is counterproductive for code generation tasks. Crucially, there is no interaction between organ and prompt effects: the improvement from organ insertion is **prompt-independent**.

The dual-organ model achieves **84.0%** on HumanEval (42/50), surpassing the base model's 80.0% (40/50) by +4 percentage points, with only 60 seconds of total training time.

Multi-seed robustness. To address the limited statistical power of a single seed, I repeat the 4-condition benchmark across 3 seeds (42, 123, 456), each sampling 50 problems from HumanEval (Table 3).

Across 3 seeds, the organ effect on the simple prompt averages +1.3pp (95% CI: [−2.0, +4.0]) and on the reasoning prompt +2.7pp (95% CI: [+0.0, +6.0]). The pooled one-sided binomial test yields p=0.38, which does not reach conventional significance at α=0.05. However, the effect direction is **consistently non-negative**: across all 6 seed×prompt combinations, 5 show improvement and 1 shows a −1 regression. This directional consistency, combined with the Reasoning-prompt CI

Table 3: Multi-seed HumanEval results (50 problems per seed, 150 total evaluations). Δ = Dual − Base.

Seed	Base(S)	Base(R)	Dual(S)	Dual(R)	Δ(S)	Δ(R)
42	40/50 (80%)	39/50 (78%)	42/50 (84%)	42/50 (84%)	+2	+3
123	44/50 (88%)	38/50 (76%)	43/50 (86%)	39/50 (78%)	−1	+1
456	40/50 (80%)	38/50 (76%)	41/50 (82%)	38/50 (76%)	+1	±0
Mean	82.7%	76.7%	84.0%	79.3%	+1.3pp	+2.7pp

lower bound of exactly 0.0, suggests a real but small effect that would require larger n to confirm statistically.

5.2 Triple Organ: Adding Japanese Capability

Extending to a triple-organ configuration (Math@5 + Code@11 + Japanese@17), I evaluate on Japanese NLP benchmarks (Table 4).

Table 4: Japanese NLP benchmark results (50 samples each, seed=42). PPL measured on 18-sample mixed corpus.

Configuration	JCom.QA	JNLI	PPL	Organs
Base Model	86.0%	**68.0%**	**5.45**	0
DUAL (Math+Code)	90.0%	62.0%	5.81	2
TRIPLE (8B Japanese)	**94.0%**	42.0%	6.48	3

A striking pattern emerges: while JCommonsenseQA *improves* monotonically with organ count (86% → 90% → 94%), JNLI *degrades* (68% → 62% → 42%). This asymmetry reveals that organ insertion selectively benefits knowledge-retrieval tasks but harms tasks requiring calibrated uncertainty.

5.3 The Confidence Bias Problem

Examining JNLI prediction distributions reveals the root cause (Table 5).

Table 5: JNLI prediction distributions reveal Confidence Bias. Ground truth: E=9, C=15, N=26.

Configuration	Acc	E	C	N	N deficit
Base Model	68.0%	14	10	26	0
Math only @5	64.0%	18	17	15	−11
Code only @11	68.0%	19	11	20	−6
DUAL (Math+Code)	62.0%	20	18	12	−14
Japanese @17 (8B)	48.0%	25	21	4	−22
TRIPLE	42.0%	34	16	**0**	−26
Math+Japanese	42.0%	27	23	**0**	−26
Code+Japanese	38.0%	33	17	**0**	−26

Key findings: (1) Each organ contributes additively to neutral suppression. (2) Combinations involving the Japanese organ completely eliminate neutral predictions (N=0). (3) The bias is *systematic*: organs push the model toward confident (entailment/contradiction) predictions, regardless of the ground truth distribution. I term this **Confidence Bias**—organ insertion disrupts the model's calibrated uncertainty, making it unable to express "I don't know."

This finding has broader implications: additive organ insertion, while preserving the host's *capabilities*, can alter its *decision calibration* in ways not captured by standard benchmarks.

Calibration metrics. To quantify this distributional shift beyond class counts, I compute class-level calibration metrics (Table 6).

Table 6: Calibration metrics for selected JNLI conditions. CCE: Class Calibration Error (mean |pred_freq − true_freq| per class). JSD: Jensen-Shannon Divergence between predicted and ground truth class distributions.

Configuration	**Acc**	**CCE**	**Brier**	**JSD**
Base Model	68.0%	0.107	0.640	0.013
DUAL (Math+Code)	62.0%	0.187	0.760	0.049
TRIPLE	42.0%	0.347	1.160	0.258
λ=0.01, L24–27	**72.0%**	**0.013**	**0.560**	**0.001**

The sweet-spot constraint (λ=0.01, L24–27) achieves the *lowest* calibration error across all metrics—even lower than the base model (CCE: 0.013 vs. 0.107; JSD: 0.001 vs. 0.013)—while simultaneously achieving the *highest* accuracy (72.0%). This demonstrates that the axis constraint does not merely restore calibration but actively *improves* it by filtering out miscalibrating organ contributions while preserving beneficial ones.

5.4 Cross-Scale Transplantation

I evaluate whether Confidence Bias persists when using Japanese organs from larger donor models with different hidden dimensions.

Table 7: Cross-scale TRIPLE organ results on JNLI and perplexity. All use standard stitching (no axis constraint).

Japanese Donor	d_D	**Scale**	**JNLI**	**E**	**C**	**N**	**PPL**
Qwen3-8B (same-family)	3584	1.0×	42.0%	34	16	0	6.48
Qwen2.5-14B	5120	1.43×	56.0%	23	18	9	6.26
Qwen2.5-32B	5120	1.43×	62.0%	21	17	12	5.91
InternLM2.5-20B	6144	1.71×	62.0%	21	17	12	5.97
Yi-1.5-34B	7168	2.00×	56.0%	19	21	10	6.15

Findings: (1) Larger donors partially restore neutral predictions (N=0 → 9–12), suggesting that cross-scale stitching layers learn a more conservative mapping. (2) Perplexity remains competitive: the 32B donor achieves 5.91 (vs. 5.45 base), demonstrating that stitching layers successfully bridge a 1.43× dimensional gap. (3) Confidence Bias is *reduced* but not eliminated—all configurations still under-predict neutral relative to ground truth (N=26).

5.4.1 Failure Case: 72B Organ (2.29×)

To probe the upper bound of cross-scale transplantation, I attempted transplanting an organ from Qwen2.5-72B-Instruct (d_D=8192, 2.29× the host dimension). This experiment failed catastrophically (Table 8).

Table 8: 72B Organ failure modes at different scale values α.

Scale α	**Quality**	**Symptom**
0.1	Degraded	Repetition loops
0.3	Collapsed	Numeric sequences ("0000...")
0.5	Collapsed	Symbol repetition ("$,$,$...")
1.0	Collapsed	Internal token leakage (Java method names, Chinese tokens)

At α=1.0, fragments of the 72B model's internal representations—Java method names (`.moveToNext`) and Chinese tokens—leak through, indicating that the stitching layers fail to correctly map the high-dimensional space. Increasing training to 3 epochs reduced loss (1.83→1.49) but worsened output quality (hallucinations). Varying insertion position (layers 5, 17, 21) did not resolve the issue.

Combined with the successful 2.0× transplantation (Yi-1.5-34B, d_D=7168), this establishes the **practical dimensional ratio limit for linear stitching at 2.0×–2.29×**.

5.5 Assistant Axis Constraint: Resolving Confidence Bias

I apply the axis-constrained training procedure (§3.6) to the TRIPLE configuration.

5.5.1 Lambda Sweep (8B organ, L22–27)

Table 9 shows the effect of constraint strength λ on the same-family 8B organ:

Table 9: Axis constraint λ sweep on TRIPLE 8B, constraint layers L22–27.

Constraint	**JNLI**	**E**	**C**	**N**	**Observation**
No constraint	42.0%	34	16	0	Full bias
Axis v1 (insertion)	38.0%	39	11	0	Worse
Axis v2 (λ=0.01)	62.0%	18	19	13	Partial recovery
Axis v2 (λ=0.05)	68.0%	15	12	23	Near base
Axis v2 (λ=0.1)	**70.0%**	14	11	25	**Exceeds base**

At λ=0.1, the axis constraint not only restores neutral predictions (N=25 vs. GT=26) but achieves **70.0% accuracy, surpassing the 68.0% base model**. The constraint effectively removes Confidence Bias while allowing beneficial organ contributions to remain.

5.5.2 Cross-Scale with Axis Constraint

I apply the optimal λ=0.1 to all cross-scale donors (Table 10).

With λ=0.1, all cross-scale configurations converge to base-level or better accuracy. The 14B, 32B, and 20B donors produce distributions *identical* to the base model, indicating the constraint fully suppresses their organ contribution on this task. The 8B and 34B donors achieve 70.0%,

Table 10: Cross-scale TRIPLE + Axis constraint (λ=0.1, L22–27) on JNLI.

Japanese Donor	**Scale**	**JNLI**	**E**	**C**	**N**
Base (no organs)	—	68.0%	14	10	26
Qwen3-8B	1.0×	**70.0%**	14	11	25
Qwen2.5-14B	1.43×	68.0%	14	10	26
Qwen2.5-32B	1.43×	68.0%	14	10	26
InternLM2.5-20B	1.71×	68.0%	14	10	26
Yi-1.5-34B	2.00×	**70.0%**	14	11	25

suggesting subtle differences in stitching geometry that allow marginal organ contribution to pass through the constraint.

5.6 Ablation: Constraint Strength × Layer Range

The cross-scale results suggest λ=0.1 may over-constrain organs. I perform a systematic ablation varying $\lambda \in \{0.001, 0.01, 0.1\}$ and constraint layers $\mathcal{C} \in \{\text{L22–27}, \text{L24–27}, \text{L26–27}\}$ on the Qwen2.5-14B donor (Table 11).

Table 11: Ablation: $\lambda\times$ constraint layers on TRIPLE 14B JNLI. GT: E=9, C=15, N=26.

λ	**Layers**	**JNLI**	**E**	**C**	**N**	**ΔBase**
Base model (no organs)		68.0%	14	10	26	—
0.1	L22–27	68.0%	14	10	26	±0
0.1	L26–27	68.0%	14	10	26	±0
0.01	L26–27	70.0%	14	12	24	+2.0
0.01	**L24–27**	**72.0%**	**14**	**15**	**21**	**+4.0**
0.01	L22–27	64.0%	17	17	16	−4.0
0.001	L26–27	64.0%	17	17	16	−4.0
0.001	L22–27	50.0%	18	27	5	−18.0
No constraint		42.0%	34	16	0	−26.0

Key findings from the ablation:

1. **Sweet spot at λ=0.01, L24–27**: This achieves the highest JNLI accuracy (72.0%, +4pp over base) with C=15 matching ground truth exactly. The constraint is strong enough to prevent Confidence Bias but permissive enough to allow beneficial organ contributions.

2. **λ=0.1 fully suppresses organs**: All λ=0.1 conditions produce distributions identical to the base model, confirming this constraint is too strong.

3. **Layer range matters**: At λ=0.01, L26–27 (70%) > L24–27 (72%) > L22–27 (64%), showing that constraining too many layers reintroduces the bias through under-constrained early layers.

4. **Clear gradient**: As constraint weakens (λ: 0.1 → 0.01 → 0.001 → 0), accuracy degrades monotonically (68 → 64 → 50 → 42%) with neutral predictions collapsing (26 → 16 → 5 → 0).

5.7 The Single-Epoch Phenomenon

My most surprising finding is that **training beyond a single epoch degrades downstream performance**, despite monotonically decreasing training loss (Table 12).

Table 12: Effect of training epochs on stitching layer quality (Code Organ, HumanEval 10 problems).

Training	Epochs	Train Loss	HumanEval
WikiText (general)	1	3.17	7/10 (70%)
Code (domain)	1	0.98	**8/10 (80%)**
Code (domain)	3	0.81	5/10 (50%)
Mixed (wiki+code)	2	1.72	7/10 (70%)

This counter-intuitive result suggests that stitching layers only need to learn a coarse representational alignment—the basic linear mapping between host and donor spaces. Additional training causes the stitching layers to overfit to the training distribution, distorting the organ's contribution in ways that harm downstream task performance.

Hypothesis. I conjecture that the optimal stitching layer approximates a near-identity transformation (for same-family transplants) or a simple rotation/scaling (for cross-family transplants). Excessive training moves the transformation away from this optimum, as the model learns to "exploit" specific patterns in the training data rather than maintaining a faithful representational bridge.

5.8 Training Data: General vs. Domain-Specific

A related finding is that **general-purpose training data (WikiText) performs comparably to or better than domain-specific data** for stitching layer training (Table 13).

Table 13: Effect of training data domain on stitching quality (Code Organ, 1 epoch).

Training Data	Train Loss	HumanEval (10)
WikiText (general)	3.17	7/10 (70%)
CodeParrot (domain)	0.98	8/10 (80%)
Mixed (50/50)	1.87	7/10 (70%)

This supports my hypothesis that stitching layers primarily learn representational alignment rather than task-specific features. The choice of training data matters less than the number of training iterations.

5.9 Ablation: Organ Placement

While the 10-problem subset does not differentiate single from dual organ configurations, the 50-problem fair comparison (Table 2) reveals a consistent +2 problem improvement from the dual organ.

Table 14: Organ configuration comparison (10-problem subset).

Configuration	HumanEval (10)
Code only (layer 11)	8/10
Math only (layer 5)	—†
Dual (Math@5 + Code@11)	8/10

†Math organ alone not evaluated on HumanEval

(code-focused benchmark).

5.10 Prompt Effect Separation

A key finding from the 4-condition comparison is that **reasoning prompts ("Think step by step") do not improve HumanEval performance**:

- Base Model: Simple 80.0% → Reasoning 78.0% (−2.0%)
- Dual Organ: Simple 84.0% → Reasoning 82.0% (−2.0%)

This suggests that reasoning prompts generate extraneous deliberation that harms implementation accuracy on code generation tasks. In contrast, the organ effect is consistent at +4.0% regardless of prompt type, indicating that **organ insertion provides a representational-level improvement independent of prompt engineering**.

5.11 Numerical Stability

Direct organ insertion (without stitching layers) causes catastrophic numerical instability for cross-family transplants:

Table 15: Numerical stability with and without stitching layers.

Organ Source	Family Match	Direct (v1)	Stitched (v2)
Qwen2.5-Coder-7B	Same (Qwen2.5)	Partial nan/inf	Stable
DeepSeek-R1-Distill-Qwen	Related	nan/inf	Stable
Qwen3-8B	Different	nan/inf	Stable

Stitching layers completely resolve numerical instability by learning to map between representational spaces, even when hidden dimensions differ (e.g., 3584 → 4096).

5.12 Computational Cost

5.13 Cross-Benchmark Evaluation

To assess organ effects beyond code generation, I evaluate five model configurations across three benchmarks: MMLU-STEM (8 subjects × 25 = 200 problems, selection-type), JCommonsenseQA (50 problems, selection-type), and MBPP (50 problems, code generation). All evaluations use seed=42 on the same problem subsets. Note that JCommonsenseQA scores here differ slightly from Table 4 due to different random samples and log-probability scoring instead of generation-based evaluation.

Three patterns emerge:

Table 16: Computational comparison of capability enhancement methods.

Method	Training Time	Data Required	Hardware	Forgetting Risk
Full Fine-tuning	Hours–Days	10K–1M+	A100 GPU	High
LoRA	1–12 Hours	1K–100K	GPU	Medium
QLoRA	30min–6h	1K–100K	Consumer GPU	Medium
This work	**30–60s**	**200**	**MacBook (MPS)**	**None by design***

*Additive architecture with frozen base model preserves original weights by construction; however, additive organ contributions can introduce soft interference (−7.5pp MMLU-STEM, §5.13) and decision calibration shifts (§5.3).

Table 17: Cross-benchmark evaluation across model configurations. Δ shows change from Base. Bold indicates best per column.

Configuration	MMLU-STEM	JComQA	MBPP
Base (Qwen2.5-7B)	58.0%	82.0%	66.0%
Dual (Math+Code)	53.5% (-4.5)	88.0% (+6.0)	**74.0%** (+8.0)
Triple (Math+Code+JP)	50.5% (-7.5)	**92.0%** (+10.0)	66.0% (±0.0)
Triple + Axis (λ=0.01)	**58.5%** (+0.5)	78.0% (-4.0)	66.0% (±0.0)
Triple + 14B Axis	56.5% (-1.5)	78.0% (-4.0)	64.0% (-2.0)

(1) Organs enhance target capabilities. The Code organ yields +8pp on MBPP (66→74%), confirming that transplanted specialist knowledge transfers to relevant benchmarks. The Japanese organ drives JCommonsenseQA from 82% to 92% in the Triple configuration—a substantial +10pp gain. These improvements validate that organ insertion provides genuine capability enhancement, not just prompt-induced effects.

(2) Multi-organ insertion degrades host knowledge. MMLU-STEM drops progressively: Base 58.0% → Dual 53.5% → Triple 50.5%. Despite the frozen base model, the additive organ contributions perturb the host's internal representations enough to degrade performance on knowledge-intensive tasks. This is a form of *soft interference*: no parameters are modified, yet the activation landscape shifts.

(3) Axis constraint recovers host knowledge but suppresses organ gains. Triple+Axis restores MMLU-STEM to baseline (58.5%) but reduces JCommonsenseQA below the base model (78% vs. 82%). The constraint effectively "neutralizes" the organ contributions along the host's identity axis, recovering knowledge retention at the cost of the very capability gains that organs provide. This reveals a fundamental tension in additive transplantation: unrestricted organ insertion maximizes target-domain gains but risks host degradation; constraining organ contributions preserves the host but limits the transplantation benefit.

6 Discussion

6.1 Why Does Single-Epoch Training Work?

I hypothesize that stitching layers serve as *representational adapters* rather than *feature learners*. Their role is to linearly map the host's hidden states into the organ's representational space (and back), not to learn new features. This mapping is relatively low-complexity—approximately a

rotation and scaling—and can be learned from minimal data. Additional training moves the stitching layers beyond this alignment role, causing them to encode distribution-specific biases that harm generalization.

This is analogous to Procrustes alignment in NLP Schönemann [1966]: aligning two embedding spaces requires only a linear transformation, which can be estimated from a small number of anchor points.

6.2 Additive vs. Replacement Transplantation

The residual connection $\mathbf{h}' = \mathbf{h} + \alpha \cdot \text{organ}(\mathbf{h})$ is fundamental to my approach. It provides:

1. **Graceful degradation**: If the organ produces unhelpful output, the residual ensures the original signal passes through.
2. **Preserved capabilities**: The base model's behavior is maintained as a default.
3. **Composability**: Multiple organs contribute additively without destructive interference.

However, my discovery of Confidence Bias (§5.3) reveals that "preserved capabilities" is more nuanced than previously understood: while the host's *knowledge* is preserved, its *decision calibration* can be disrupted. The residual connection guarantees capability preservation but not distributional preservation.

6.3 Understanding Confidence Bias

I hypothesize that Confidence Bias arises because organ MLPs, having been trained in specialist models, encode a prior toward confident predictions. In natural language inference, the base model's neutral predictions rely on a delicate balance of activation magnitudes—the model must "decide not to decide." Organ contributions, however small (α=0.1), perturb activations in directions that favor entailment or contradiction over neutral. This effect is additive: each organ independently biases the distribution, explaining why multi-organ configurations exhibit stronger bias (Table 5).

The Assistant Axis constraint addresses this by projecting out the component of organ influence along the model's identity direction. Critically, this does not remove organ contributions entirely—it only constrains the dimension that encodes the model's "being itself" (calibrated, balanced response behavior). Orthogonal contributions from the organ pass through unconstrained, explaining why the optimal setting (λ=0.01, L24–27) *surpasses* the base model: the organ provides genuine task-relevant signal that the constraint allows to flow through.

6.4 The Task-Ability Mismatch

An important observation from my JNLI experiments is that the Japanese organ does not improve JNLI accuracy in isolation (48.0% vs. 68.0% base). This is not a failure of transplantation but a **task-ability mismatch**: JNLI primarily tests logical reasoning over Japanese text, not Japanese language competence. The Japanese organ provides linguistic capability (vocabulary, grammar, fluency) but not inferential capability. This distinction has implications for organ selection—the appropriate organ for a task must match the *cognitive ability* the task demands, not just the *language* or *domain*.

6.5 Democratization of Model Enhancement

Perhaps the most significant practical implication is accessibility. My method enables:

- A researcher with a laptop to enhance a 7B model in under a minute
- Plug-and-play capability modules that can be shared as small files (∼100MB per organ)
- Cross-architecture transplantation: organs from 14B–34B models can be transplanted into a 7B host via learned dimensional bridges
- Experimentation with organ combinations without expensive retraining

This lowers the barrier to model customization from "requires a GPU cluster" to "requires a laptop."

6.6 Limitations

1. **Evaluation scale**: My main code generation results are based on 50 HumanEval problems per seed. A 3-seed replication (150 total evaluations) shows a consistent positive organ effect (+1.3pp simple, +2.7pp reasoning) but the pooled binomial test remains non-significant (p=0.38). Cross-benchmark evaluation (§5.13) on MMLU-STEM, JCommonsenseQA, and MBPP confirms these trends with larger effect sizes (+8pp MBPP for Dual, +10pp JComQA for Triple), but all evaluations use a single seed (42) with 50–200 problems per benchmark.
2. **Single host model**: I evaluate only on Qwen2.5-7B-Instruct. Generalization to other architectures (Llama, Mistral) remains to be verified.
3. **JNLI as primary NLI benchmark**: The 50-sample subset provides limited statistical power. While the Confidence Bias pattern is consistent across all conditions, individual accuracy numbers have an inherent noise margin.
4. **Axis constraint generality**: The assistant axis is model-specific and requires a one-time extraction step (∼70 minutes). Whether the same axis effectively constrains across different task types is untested.
5. **No automatic metric for Japanese quality**: I was unable to evaluate Japanese generation quality (fluency, naturalness) due to the lack of reliable automatic metrics. Grammatically correct Japanese does not equate to natural Japanese, and most native speakers produce grammatically imperfect but natural text—a distinction no current metric captures.
6. **Inference overhead**: Each organ adds a forward pass through additional MLP layers, increasing inference latency by approximately 10–15% per organ.

6.7 Future Work

1. **Automatic constraint tuning**: Learning optimal λ and layer range per task, potentially via a small validation set.
2. **Organ library**: Building a repository of sharable organs with documented Confidence Bias profiles.

3. **Task-aware organ routing**: The cross-benchmark evaluation (§5.13) reveals that applying all organs uniformly causes soft interference on unrelated tasks. A learned gating mechanism—selecting which organs to activate based on the input, analogous to MoE routing—could preserve organ gains on target tasks while avoiding host degradation on others. This addresses the fundamental tension between capability enhancement and knowledge preservation that the axis constraint trades off rather than resolves.

4. **Scaling laws for transplantation**: Understanding how the organ effect scales with host and donor model sizes.

5. **Japanese generation evaluation**: Developing human-in-the-loop or learned metrics for Japanese text naturalness, addressing the fundamental limitation of current automatic evaluation.

7 Conclusion

I presented Hamburger Transplantation, a lightweight and accessible method for enhancing LLM capabilities through additive organ insertion. By extracting frozen MLP layers from specialist models and connecting them to a host model via thin stitching layers, I achieve measurable capability improvements with just 30–60 seconds of training on consumer hardware.

My investigation revealed both the promise and the pitfalls of additive transplantation. On the positive side: dual-organ insertion yields a consistent +4pp improvement on HumanEval, the Code organ provides +8pp on MBPP, and the Japanese organ drives +10pp on JCommonsenseQA—confirming genuine cross-benchmark capability transfer. Cross-scale transplantation enables organs from models up to 2× the host's hidden dimension.

On the cautionary side, I identified two distinct failure modes. First, **Confidence Bias**: organ insertion systematically suppresses neutral predictions in NLI tasks, collapsing a 3-class distribution into 2 classes. Second, **soft interference**: multi-organ insertion degrades host knowledge on MMLU-STEM by up to −7.5pp despite frozen base parameters, demonstrating that additive contributions can perturb the host's activation landscape without modifying any weights.

My proposed solution—Assistant Axis Constraint Training—resolves Confidence Bias and recovers host knowledge (MMLU-STEM restored to baseline), but at a cost: the constraint suppresses the very organ gains it was designed to preserve, reducing JCommonsenseQA below the base model (78% vs. 82%). The ablation study identifies a sweet spot (λ=0.01, L24–27) that surpasses the base model on JNLI (72.0% vs. 68.0%), but the cross-benchmark evaluation reveals this as task-specific rather than universal.

The broader lesson is that additive transplantation involves not one but two preservation challenges: *capability preservation* (maintained by residual connections) and *distributional preservation* (disrupted by organ contributions). The axis constraint trades one form of preservation for another. Resolving this fundamental tension—perhaps through input-conditional organ gating—remains the central open problem for modular neural architectures.

All experiments were conducted on a single MacBook Pro (M4 Pro, 24GB), demonstrating that meaningful research on neural organ transplantation is accessible without GPU clusters.

References

Bansal, Y., Nakkiran, P., and Barak, B. Revisiting Model Stitching to Compare Neural Representations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H.P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*, 2021.

Csiszárik, A., Kőrösi-Szabó, P., Matszangosz, Á.K., Papp, G., and Varga, D. Similarity and Matching of Neural Network Representations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized Language Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Fedus, W., Zoph, B., and Shazeer, N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025.

Kurihara, K., Kawahara, D., and Shibata, T. JGLUE: Japanese General Language Understanding Evaluation. *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC)*, 2022.

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *International Conference on Learning Representations (ICLR)*, 2022.

Hui, B., Yang, J., Cui, Z., Yang, J., Liu, D., Zhang, L., Liu, T., Zhang, J., Yu, B., Dang, K., et al. Qwen2.5-Coder Technical Report. *arXiv preprint arXiv:2409.12186*, 2024.

Ilharco, G., Ribeiro, M.T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing Models with Task Arithmetic. *International Conference on Learning Representations (ICLR)*, 2023.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Vinyals, O., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Komatsuzaki, A., Puigcerver, J., Lee-Thorp, J., Ruiz, C.R., Mustafa, B., Ainslie, J., Tay, Y., Dehghani, M., and Houlsby, N. Sparse Upcycling: Training Mixture-of-Experts from Dense Checkpoints. *International Conference on Learning Representations (ICLR)*, 2023.

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C.F., Cheng, K.-T., and Chen, M.-H. DoRA: Weight-Decomposed Low-Rank Adaptation. *International Conference on Machine Learning (ICML)*, 2024.

Qwen Team. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024.

Schönemann, P.H. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

Shazeer, N. GLU Variants Improve Transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *International Conference on Learning Representations (ICLR)*, 2017.

Wortsman, M., Ilharco, G., Gadre, S.Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A.S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *International Conference on Machine Learning (ICML)*, 2022.

Yadav, P., Tam, D., Choshen, L., Raffel, C., and Bansal, M. TIES-Merging: Resolving Interference When Merging Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Al-Zuraiqi, A. Neural Organ Transplantation (NOT): Checkpoint-Based Modular Adaptation for Transformer Models. *arXiv preprint arXiv:2601.13580*, 2026.

Yu, L., Yu, B., Yu, H., Huang, F., and Li, Y. Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch. *International Conference on Machine Learning (ICML)*, 2024.

Tenney, I., Das, D., and Pavlick, E. BERT Rediscovers the Classical NLP Pipeline. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Jawahar, G., Sagot, B., and Seddah, D. What Does BERT Learn about the Structure of Language? *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Men, X., Xu, M., Zhang, Q., Wang, B., Lin, H., Lu, Y., Han, X., and Chen, W. ShortGPT: Layers in Large Language Models are More Redundant Than You Expect. *arXiv preprint arXiv:2403.03853*, 2024.

A Detailed Problem-Level Results

Table 18: Problem-level results on HumanEval 10-problem subset (seed=42).

Task ID	**Base**	**Code 1ep**	**Dual+Reason**	**Difficulty**
HumanEval/163	✓	×	×	Tricky ("even digits")
HumanEval/28	✓	✓	✓	Standard
HumanEval/6	✓	✓	✓	Standard
HumanEval/70	✓	✓	✓	Standard
HumanEval/62	✓	✓	✓	Standard
HumanEval/57	✓	✓	✓	Standard
HumanEval/35	✓	✓	✓	Standard
HumanEval/26	✓	×	✓	Tricky (semantics)
HumanEval/139	✓	✓	✓	Standard
HumanEval/22	×	✓	✓	Standard
Total	**8/10**	**8/10**	**9/10**	

B Reproduction Guide

All experiments can be reproduced on a consumer laptop with ≥24GB RAM:

```
# Step 1: Install
pip install -e .

# Step 2: Download weights
python scripts/download_weights.py --repo your-username/hamburger-transplant-weights

# Step 3: Train stitching layers (30s each)
python scripts/train_math_stitching.py
python scripts/train_code_stitching.py

# Step 4: Fair 4-condition benchmark (~80 min)
python scripts/benchmark_fair.py --num-problems 50
```

Code and organ weights are available at: `https://github.com/tatsuru-okada-business/hamburger-transpl`